Service integration in distributed control systems: an approach based on fusion of mereologies — Source link ↗

Richard Dapoigny, Nacima Mellal, Eric Benoit, L. Foulloy

Related papers:

- Application Of Dynamic Knowledge In The Design Of Distributed Operating Systems

- Distributed Active Objects -- A Systemic Approach to Distributed Mobile Applications

- An Application Framework for Loosely Coupled Networked Cyber-Physical Systems

- Types and their management in open distributed systems

- Deployment Modelling and Synthesis in a Component Model for Distributed Embedded Systems

Service integration in distributed control systems: an approach based on fusion of mereologies

Richard Dapoigny, Nacima Mellal, Eric Benoit, Laurent Foulloy

HAL Id: hal-00724281

https://hal.archives-ouvertes.fr/hal-00724281

Submitted on 20 Aug 2012

Service Integration in Distributed Control Systems: an approach based on fusion of mereologies

Richard Dapoigny*, Nacima Mellal*, Eric Benoit* and Laurent Foulloy*

*Université de Savoie, ESIA

Laboratoire d'Informatique, Systèmes,

Traitement de l'Information et de la Connaissance

B.P. 806, F-74016 ANNECY Cedex, France

Phone: +33 450 096529 Fax: +33 450 096559

Email: richard.dapoigny@univ-savoie.fr

Abstract— The objective of the present paper is to provide a methodology where the functional characteristics of a distributed engineering system can be obtained by merging domain-dependent knowledge at run-time. We focus on distributed control systems where computing nodes are related to the physical environment in which they operate via sensors/actuators. The knowledge representation is formally expressed with a mereological approach where a structural mereology describes the physical environment and a functional mereology identifies available engineering goals for each computing node. During the design step, a mechanism based on Formal Concept Analysis (FCA) allows to generate the resulting goal mereology. The concept of goal is refined with sub-concepts in the multilevel structure. Because computing nodes depend on each other for goals to be achieved, an agent-based method is proposed to establish dynamically the dependencies among distributed nodes. This method is centered on a fusion mechanism involving the functional mereologies of appropriate nodes. We use an example from an open-channel hydraulic system controlling the water level to motivate and illustrate the model. Although it is limited to the engineering systems, this approach can be reused in related domains where the goal representation can be expressed as a triple including an action, a role and a physical entity.

I. INTRODUCTION

Engineering services are a new paradigm for building distributed control applications. In this article, given a global service that a control unit should achieve, we propose first to formalize and to subdivide the service into a hierarchy of elementary services. Then we intend to identify on a formal basis (i.e., a fusion process) dependent services (in a network of control units) which are required to achieve the global service. We consider a network composed of intelligent nodes which are either sensing their physical environment or/and acting upon it and which are able to exchange information with each other [2][3]. These intelligent nodes arranged with a computing part and I/O capabilities, will be referred to as Intelligent Instruments (II) in the following. To get II to become autonomous nodes, the basic idea consists in the decomposition of a global service into a hierarchy of sub-services which can be distributed over the network of II. As a consequence, dependence relations emerge between distributed services. The key element to dynamically extract these dependencies holds in the knowledge representation of the service concept. A previous work [14] has suggested a representation centered on a teleological approach where each service is related to the goal concept.

In order to automate the goal decomposition and the goal fusion processes, we suggest a two-phase sequence. In a first step, we propose to identify the goal hierarchy through formal concept analysis and a mereological approach. In a second step, inter-dependencies between distributed hierarchies of goals are highlighted through a fusion process. All theses steps involve a mereological framework. The main purpose of mereology [1] is to identify the parts that particular entities can exhibit and the way those parts are inter-related [18]. Broadly speaking, the ontological representation provides a robust basis for inter-operability and information sharing. In the ontological approach, the choice of a formal modelling basis is either set theory or mereology. Some authors [19] consider that set theory which is an abstraction is unable to describe reality which is not an abstraction. Moreover, the relations between entities are not set-theoretic in nature [17], therefore a mereological approach is investigated. While the state variables are semantically described with a physical role and a physical entity, each goal (or sub-goal) is a composite concept which is made of atomic concepts such as action, physical role and physical entity. We introduce successively universal goals composed with action and physical role, which are generic and re-usable, and particular goals composed of action, physical role and physical entity which are related to the application. Particular goals are derived either from user inputs or from universal goals extracted from libraries. For each instrument, the goal mereology is deduced from rules applied to the user inputs through an intelligent GUI. At run-time, an agent-based implementation operates the fusion of appropriate mereologies to solve dynamically dependencies between distributed goals (i.e., distributed services).

In a second section, we discuss the selection of the relevant concepts involved in II. In the third section, we introduce the real-world application which will serve as a support for a clear understanding of the fusion process. The fourth section recalls the principles underlying the structural mereology. The fifth section is dedicated to the goal modelling where the concepts of universal and particular goals are introduced and describes the conceptual goal-subgoal hierarchy dedicated to distributed

II. Rough outlines for the implementation of the fusion process are proposed. Related work are discussed in the last section.

II. KNOWLEDGE REPRESENTATION FOR PHYSICAL PROCESSES IN DISTRIBUTED CONTROL SYSTEMS

Important ideas that have emerged in the formalization of limited domains of knowledge and reasoning, are the introduction of an explicit ontology [4] and an explicit representation of context [6]. These results have entailed a basic idea, i.e., the knowledge representation is tailored to a representational knowledge base which structures and relates words of the engineering language. In related areas such as Process Knowledge Bases [5], the authors suggest that complete information about an engineering or computer information system is expressed with three types of knowledge (i.e., structural, functional and behavioral knowledges). We adopt this representation for distributed II where the knowledge about the system will be divided into three parts and their related knowledge:

- structural knowledge – knowledge about physical objects and their physical relations
- functional knowledge – knowledge about the purpose and the way the physical role of objects are used.
- behavioral knowledge – knowledge about the changes in the environment and relations between states and events.

The structural knowledge describes the physical environment of sensors/actuators and relies on theories applied to engineering processes [7]. For physical processes, this knowledge is centered on two concepts, i.e., the physical entity and the physical role or quantity [8]. In II, the functional knowledge, through its teleological part, is represented by the concept of goal (sub-goal), while the dynamic part is related to the concept of action [14]. In order to capture the various objectives that the engineering system should achieve, goals must represent different abstraction levels, and a hierarchical representation such as mereology is a well-suited formalism to express these levels. The notion of universal goal relating an action verb and a physical role is introduced and, by adding an extensional item (i.e., the physical entity) and domain-based rules to the universal goal, the particular goals are defined. In order to allow reuse and hierarchical conceptual clustering of goals, a goal mereology is derived. This mereology is elicited during the design step from the interaction with the user. At run-time, each situated agent is responsible from the gathering of dependent sub-goals in some of the II. A dedicated rule-based process extracts the appropriate sub-goals and their mereological wholes.

III. AN EXAMPLE OF DISTRIBUTED CONTROL APPLICATION

To illustrate the fusion process between distributed instruments, we propose a case study which concerns an open-channel hydraulic system controlled with (at least) two intelligent instruments as shown in figure 1. The instruments are connected with a CAN network. Each II used in the open-channel irrigation channel is located near a water gate. Each instrument performs two pressure measurements from
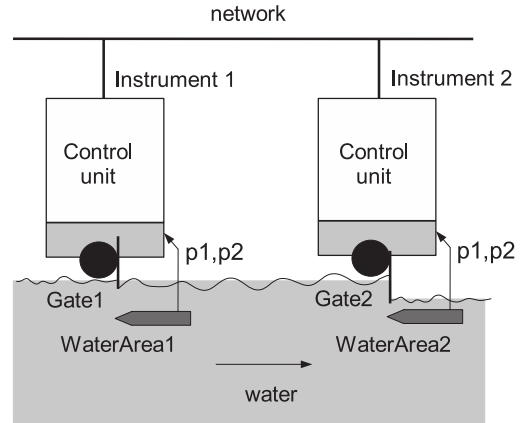

Fig. 1.   The canal application with two Intelligent control nodes

a Pitot tube and is able to react accordingly and to modify the gate position with the help of a brushless motor. The Pitot tube allows two pressure measurements, a static pressure in spatial areas denoted $SFArea11$ or $SFArea21$, and a dynamic pressure measurement in areas denoted $DFArea12$ or $DFArea22$. The local area surrounding the two previous is respectively referred as $WaterArea1$ and $WaterArea2$ for the first and the second instrument. Goals are user-defined through the graphical GUI. All functions handle variables whose semantic contents are extracted from the structural mereology. A global goal is downloaded in the second instrument, where the objective is to achieve this intended goal. For this purpose, the situated agent must first discover which services on another instrument are required and then produce the appropriate goal hierarchy with relevant goals to achieve. Notice that the fusion process is not intended to merge the whole mereologies from different instruments, but only to merge the relevant parts of the local mereologies that are required to achieve the global goal.

IV. THE STRUCTURAL MEREOLOGY

The physical environment is first described through a GUI during the design step to identify the physical entities which are hierarchically and formally related through the mereological logic. To describe the physical behavior of physical entities, we must express the way these entities interact. As suggested in [7] the physical interactions are the result of energetic physical processes that occur in physical entities. Whatever two entities are able to exchange energy, they are said to be connected. Therefore, the mereology is extended with a topology where connections highlight the energy paths between physical entities. This approach extracts in a local database, energy paths stretching between computing nodes in the physical environment. The corresponding mereo-topology is reported on figure 2.
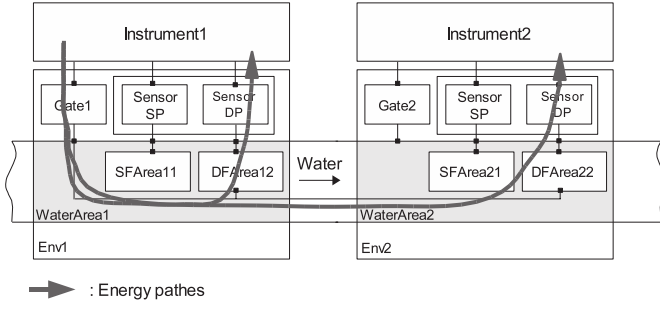
Fig. 2. The structural mereo-topology for the canal application (2 II)

V. THE GOAL HIERARCHY

A. Sub-goal representation

In the context of intelligent instruments, any functional concept will be described by a (sub-)goal definition[1] which is related to the intensional aspect of function [9] and some possible actions (at least one) in order to fulfill the intended (sub-)goal [10][11]. Representation of intended goals as "to do X" has been used by several researchers [12][13] and we extend that textual definition. We have made an explicit use of the mereology for the representation level and Formal Concept Analysis (FCA) for the concept level. The sub-goal modelling requires first to describe sub-goal representation (i.e., data structures), and secondly to define how these concepts are related.

Definition 1: An universal sub-goal item belongs to the relation $J \subseteq R \times A$, where A, and R are the respective finite sets associated with the respective sorts of verbs describing elementary actions and the sort of physical roles (physical quantities or control/monitoring roles).
The potential action is added to the physical role which describes the intensional part of the universal sub-goal [2]. A small number of pre-defined generic verbs which describe a limited set of appropriate elementary actions [14] related with standard roles, creates elementary sub-goals which can be joined to form more complex goals within conceptual hierarchies. Examples of elementary universal sub-goals:

$g1 = (pressure, to\_acquire)$
$g2 = (position, to\_move)$
$g3 = (speed, to\_compute)$

The set of objects, i.e., physical entities extracted from the structural mereology is related to the objects properties, i.e., their physical roles. A set of objects and their attributes represents relevant information and knowledge about the environment. Therefore, one can introduce a variable concept relating physical entities with their physical role (physical quantity).
Examples of physical variables concepts:

$(\{SFArea1, DFArea1\}, \{Pressure\})$
$(\{WaterArea1\}, \{level, speed\})$

---

[1]assuming the teleological interpretation of functions
[2]This description corresponds formally to that of a context item

Similarly, the universal goal definition is extended with an extensional part (i.e., the physical entity) to generate particular sub-goals from physical variables and universal sub-goals contexts.

Definition 2: A particular sub-goal concept g_i is a triple such as :

$$g_i = (a_i, \bigcup_j r_{ij}, \bigcup_k \varphi_{ik}) \qquad (1)$$

where a_i denotes an elementary action, $\bigcup_j r_{ij}$, a set of physical roles (at least one) and $\bigcup_k \varphi_{ik}$, a set of physical entities (at least one) concerned by the action a_i.
Some examples of particular sub-goals :

$g_1 = (\{to\_acquire\}, \{pressure\}, \{SFArea1\})$
$g_3 = (\{to\_compute\}, \{speed, level\}, \{WaterArea1\})$

Ontologies can be analyzed with FCA techniques [15] and this work has been extended with mereology. In FCA, each concept is expressed as a unit of thought comprising two parts, its extension and its intension. FCA produces a conceptual hierarchy of the domain by exploring all possible formal concepts for which relationships between properties and objects hold [16]. The resulting concept lattice, also known as Galois Lattice, can be considered as a semantic net providing both a conceptual hierarchy of objects and a representation of possible implications between properties.

B. The conceptual goal hierarchy

The goal mereology is derived from the subsumption hierarchy of conceptual scales where the many-level architecture of conceptual scales [20] is extended taking into consideration the mereological nature of the extents. Higher level scales which relates scales on a higher level of abstraction provide information about hierarchy and help to derive a hierarchy like the mereology. Considering the particular sub-goals the particular goals and modes corresponding to the user intents, the ontological nature of the extents (i.e., the physical entities) and some basic assumptions, one can automatically produce the relevant instrument functional context. This context is required to produce the final concept lattice from which the functional mereology is extracted.
Let us consider the formal context $C = (\Phi, G, F)$, where G denotes the set of relevant particular sub-goals and $F \subseteq \Phi \times G$, a binary relation which holds between physical entities and particular sub-goals. As suggested in [20], the set of sub-goals is extended with hierarchical conceptual scales such as the intent includes sub-goals, goals (i.e., services) and the instrument scale (highest level). Higher level scales define a partially ordered set $(\mathcal{G}, \leqslant)$ provided that the set G contains exactly the minimal elements of $\mathcal{G}$. Hierarchical conceptual scales are filled according to information input by the user concerning goals definitions. Then the conceptual hierarchy highlights required inter-relations between concepts. For the open-channel irrigation canal, three services are introduced by the user, related to their sub-goals:

$G_1 = (\{to\_measure\}, \{speed, level\}, \{WaterArea1\})$
$G_2 = (\{to\_control\}, \{speed\}, \{WaterArea1\})$

TABLE I

INSTRUMENT FUNCTIONAL CONTEXT FOR THE OPEN-CHANNEL IRRIGATION CANAL

$\mathcal{F}$	g_1	g_2	g_3	g_4	g_5	g_6	g_7	G_1	G_2	G_3	II_1
(pressure, SFArea1)	x							x	x		x
(pressure, DFArea1)		x						x	x		x
(speed, WaterArea1)			x	x	x			x	x		x
(level, WaterArea1)			x	x				x	x		x
(position, Gate1)						x			x	x	x
(speed, ?)					x		x		x		

$G_3 = (\{to\_manuallyMove\}, \{position\}, \{Gate1\})$

$g_1 = (\{to\_acquire\}, \{pressure\}, \{SFArea1\})$

$g_2 = (\{to\_acquire\}, \{pressure\}, \{DFArea1\})$

$g_3 = (\{to\_compute\}, \{speed, level\}, \{WaterArea1\})$

$g_4 = (\{to\_send\}, \{speed, level\}, \{WaterArea1\})$

$g_5 = (\{to\_compare\}, \{speed\}, \{WaterArea1, ?\})$

$g_6 = (\{to\_move\}, \{position\}, \{Gate1\})$

$g_7 = (\{to\_receive\}, \{speed\}, \{?\})$

From this information, the functional context is built, where goals and sub-goals denote the intensional part and variables conceptual pairs denotes the extensional part. The instrument level expresses the functional knowledge about the instruments' goals and sub-goals that are achievable with the help of local variables. This level encapsulates all locally-achievable goals.

$II_1 = (\{to\_control\}, \{speed, level\}, \{Env1\})$

From the users' point of view, two initial levels are defined while the formal concept analysis enhanced by a mereological translation may result in an arbitrary number of intermediary levels. The functional context results in a concept lattice which is finally transformed in a partial order by some elementary rules:

rule 1: to reduce the labelling [21] which provides that each object and each attribute is entered once in the lattice.

rule 2: emphasize differences between wholes and parts through identical variables use i.e., $\{G_3, g_6\}$ will result in $P(g_6, G_3)$

rule 3: extract common parts between set items, i.e., overlap relations.

rule 4: create for each node a concept labelled with the intension of the lattice node [22]

rule 5: remove the bottom element

In the reduced hierarchy, goals are mereologically ordered according to their physical entity extent[3] and generate the Var-mereology of the instrument. In this mereology, a first hierarchy of goals reflects the sub-goals' use of variables until the node Instr. We notice that the goal G_2 subsumes the instrument node, which corresponds to the fact that G_2 requires external information whereas the instrument only deals with its local structural mereology. This entails that external information will be necessary at run-time. The common node g_3, g_4, g_5 points out that these goals share a common variable concept, i.e., $(\{speed\}, \{WaterArea1\})$. As a consequence,

---

[3] other classifications are possible, using different object types.

goals g_3, g_4, g_5 overlap according to the Var-mereology. The reduced lattice is reported on figure 3 and the resulting mereology is sketched in figure 4 (sub-goals overlaps are not detailed for clarification). The current environment developed
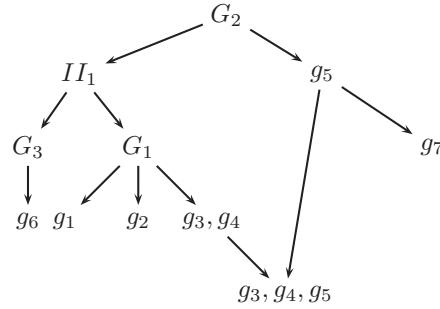

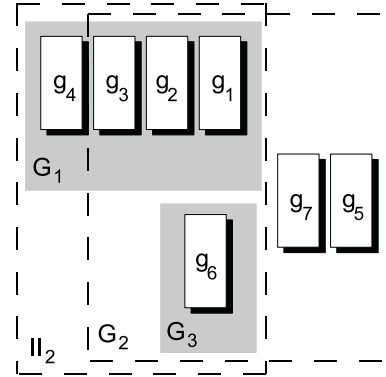Fig. 3. The reduced lattice for the open-channel irrigation canal

Fig. 4. The mereology for the instrument 2

with an open full Java version including XProlog code and libraries. The meta-description of the instrument is shared with other applications II at run-time.

C. The goal fusion process

From the previous mereology, goals are hierarchically ordered according to their amount of required variables. Goals which contain a question mark as physical entity need external variables. Therefore external goals are needed which are able to capture the missing resources, i.e., the physical entity. Goals (or sub-goals) with missing entities are referred to as non-reified goals. In fact, the missing resource

is shared with an external goal, and the challenge boils down to an identification problem with particular goals. As particular goals are mereological entities, the shared entities are translated in overlap operations. As a consequence, the key element of the algorithm is the identification process of overlapping goals or sub-goals with constraints. Following the identification process, related goals (i.e., services) are extracted and their restricted fusion[4] with the local goal is achieved. A multi-agent system implements the algorithm sketched in figure 5 on the basis of a message passing protocol between instruments. To illustrate the fusion process, we
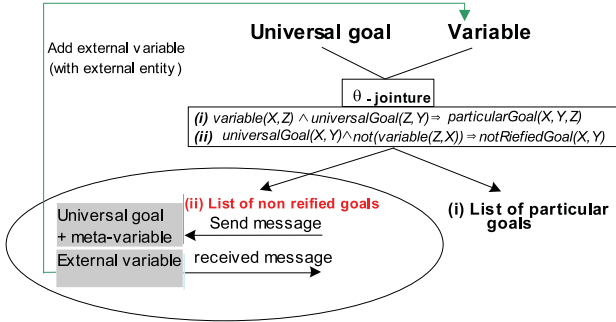

Fig. 5.   The fusion process algorithm

propose a scenario using the physical environment described in section 3.

Initial problem: A global goal, namely G_2, is assigned to instrument 2. In other words, the challenge is to regulate the water level from II2. To satisfy that goal, a sound hierarchy is proposed (see section 4) which highlights that external goals from II2 are required. The objective of the present section is to discover which external goal(s) is (are) required and to add dynamically this knowledge to the database of II2.

step 1: *Location of related instruments.* The situated agent in II2 looks for the instrument(s) with which there exists an energy path extracted from the Structural mereology. The answer is II1.

step 2: *Discovering relevant local subgoals.* Exploring the local mereology in II2, the agent notices that G_2 overlaps II2 (see figure 4), which means that G_2 requires external sub-goals to complete. In this case, sub-goals g_7 and g_5 needs external data to be achieved.

step 3: *External sub-goals extraction.* Rule: the local sub-goals (from step 2) having missing physical entities overlap the external sub-goals producing variables on the network with the same physical role and which belong to a related II (step 1). The resulting relation is here, $O(g_5, g_4')$, as this sub-goals share the semantic variable ($\{speed\}, \{WaterArea2\}$).

---

[4] it means that only the required part of the external mereology is merged with the local one.

step 4: *External service recovering.* The high-level goals are deduced from overlapping and part-of axioms as follows:

$$O(g_5, g_4') \wedge P(g_5, G_2) \wedge P(g_4', G_1') \rightarrow O(G_2, G_1').$$

This result shows that service G_2 on instrument 2 needs the service G_1' on instrument 1 in order to be achieved. In concrete terms, II1 performs two pressure measurements, computes the resulting water speeds and levels and sends the results to II2 (i.e., G_1'). Instrument 2 activates goal G_2, that is, it performs two pressure measurements, computes the resulting water speeds and levels, compares them to the results from II1 and finally, reacts on gate 2 according to the comparison of distributed measurements. The summarized fusion of mereologies is reported on figure 6. Notice that, in order to clarify the explanation, only relevant overlaps are detailed on the figure.
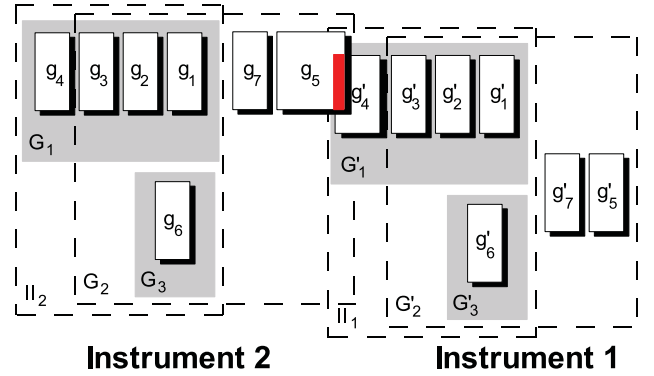

Fig. 6.   The fusion process for mereologies of instruments 1 and 2

VI. RELATED WORK

In process Control Systems Design, dedicated tools such as described in [26] gathers and structures the requirements for Process Control Systems in a two-phases method. First goals are identified, classified and then combined together to deliver a complete goal model. In the second phase, the goal model is translated into B specifications [28] through a state-transition diagram. Unfortunately, the goal description is rather poor first due to a lack of formalization to describe goals relations, secondly because there are no distinctions within the goal model between intended goals and their way of achievement and finally, the model is not suitable to support distributed applications involving dynamic goal knowledge. In [23] an acquisition assistant is proposed which operationalizes the goals with constraints. The structure of goals does not allow further reasoning and no automated support is provided. More recently, in [24] goals are represented by verbs with parameters, each of them playing a special role such as target entities affected by the goal, resources needed for the goal achievement, ... Some tools are based on temporal logic and offer refinement techniques to link goals [25]. For more general frameworks, [27] describes a logic of goals based on

their relationship types, but goals are only represented with a label, and the reasoning is elicited from their relations only. In this article, we have emphasized a conceptual representation of goals that serves as a basis for mereologic fusion.

VII. CONCLUSION

Intelligent Instruments are obviously intended for physicians or engineers. The proposed automatization of appropriate teleologic information is crucial to ensure the proper function of distributed control systems. In addition, there is a growing need for a structured knowledge base to allow both reuse and distributed reasoning at run-time. The bottom-up approach classifies concept-subconcept relations with conceptual scales and allows to obtain automatically the resulting mereology of goal-subgoals that holds for a given instrument. With this representation agents may reason about actions at multiple levels of granularity, provided that the consistency between goal levels is achieved by mereological axioms. Moreover, the goal hierarchy provides vertical traceability from high-level concerns to low-level technical details. Goal formalization allows refinements to be proved sound and complete. Due to the modular knowledge representation, complex applications can be divided into smaller sub-applications across several instruments which interact each other to exchange appropriate information in so-called fusion processes. Unfortunately, this model is not complete because it doesn't take into account the causal nature of engineering functions. As a consequence, additional logic is needed for composing partial-order planning and hierarchical task network planning. Future efforts include such a formal modelling for reasoning about interactions with the plan and resolving conflicts.

REFERENCES

[1] Simons, P.: *Parts: a Study in Ontology*, Oxford University Press 1987

[2] Kopetz, H., Holzmann, M., Elmenreich, W.: *A Universal Smart Transducer Interface : TTP/A*, International Journal of Computer System Science and Engineering. **16**(2) 2001

[3] Bouras A., Staroswiecki M.: *How can Intelligent Instruments Interoperate in an Application Framework ? A Mechanism for Taking into Account Operating Constraints*, IFAC Symposium on Intelligent Components and Instruments for Control Applications (SICICA). 1997 465-472

[4] Fikes R.: *Ontologies: What are they, and where's the research?*, Principles of Knowledge Representation and Reasoning. 1996 652–654

[5] Dooley K., Skilton P., Anderson J.: *Process Knowledge Bases: Facilitating Reasoning through Cause and Effect Thinking*, Human Systems Management, **17** (4), 1998 281-298

[6] Guha R.V.: *Contexts: a Formalization and some Applications*, Stanford Computer Science Departement **STAN-CS-91-1399** 1991

[7] Borst, P., Akkermans H., Pos A., Top J.: *The PhysSys Ontology for Physical Systems*, 9th international Workshop on Qualitative Reasoning. 1995 11–21

[8] Dapoigny R., Benoit E., Foulloy L.: *Ontology Implementation for Knowledge Representation in Intelligent Instruments*, 2th IEEE Symposium on Signal Processing and Information Technology 2002 17–21

[9] Lind M.: *Modeling Goals and Functions of Complex Industrial Plant*, Journal of Applied Artificial Intelligence **8** 1994 259–283

[10] Hertzberg J., Thiebaux S.: *Turning an Action Formalism into a Planner: a case Study*, Journal of Logic and Computation. **4** 1994 617-654

[11] Lifschitz V.: *A Theory of Actions*, Procs. of the 10th International Joint Conference on Artificial Intelligence 1993 432-437

[12] Umeda Y., and al.: *Supporting conceptual design based on the function-behavior-state modeler*, Artificial Intell. for Engineering Design, Analysis and Manufacturing. **10** (4) 1996 275–288

[13] Qian L., Gero J.S.: *Function-behavior-structure paths and their role in analogy-based design*, Artificial Intell. for Engineering Design, Analysis and Manufacturing. **10** (4) 1996 289–312

[14] Dapoigny R., Benoit E., Foulloy L.: *Functional Ontology for Intelligent Instruments*, Foundations of Intelligent Systems. LNAI 2871 (Springer) 2003 88–92

[15] Cimiano P., Hotho A., Stumme G., Tane J.: *Conceptual knowledge processing with formal concept analysis and ontologies*, ICFCA. LNAI 2961. 2004 189–207

[16] Wille R.: *Restructuring lattice theory: an approach based on hierarchies of concepts*, Ordered sets. Reidel, Dordrecht-Boston. 1982 445–470

[17] Degen, W., Heller, B., Herre, H. and Smith, B. *GOL: A general ontology language*, in B.S. Welty, Christopher, ed., Procs. of the 2nd International Conference on Formal Ontology in Information Systems, ACM Press, New York, 2001 34–45.

[18] Smith, B.: *On substances, accidents and universals: In defense of a constituent ontology*, Philosophical Paper **27**, 1997 105–127.

[19] Smith, B.: *Basic concepts of formal ontology*, in N. Guarino, ed., Formal Ontology in Information Systems, IOS Press, Amsterdam, 1998 19–28.

[20] Stumme G.: *Hierarchies of conceptual scales.* Procs. of the Workshop on Knowledge Acquisition, Modeling and Managment (KAW'99). **2** 1999 78–95

[21] Ganter B., Wille R.: *Formal concept analysis - mathematical foundations* 1999 Springer.

[22] Cimiano P., Staab S., Tane J.: *Deriving concept hierarchies from text by smooth formal concept analysis.* Procs. of the GI Workshop LLWA 2003

[23] Dardenne A., van Lamsweerde A., Fickas S.: *Goal-directed requirements acquisition.* Science of computer programming. **20** 1993 3–50

[24] Rolland C., Souveyet C., Ben Achour C.: *Guiding goal modelling using scenarios.* IEEE Trans. on software eng. 1998 1055–1071

[25] Letier E.: *Reasoning about agents in goal-oriented requirements engineering.* Doct. dissertation. University of Louvain. 2001

[26] El-Maddah I., Maibaum T.: *Goal-oriented requirements analysis for process control systems design.* Procs. of the int. conf. on formal methods and models for co-design. 2003

[27] Giorgini P., Nicchiarelli E., Mylopoulos J., Sebastiani R.: *Reasoning with goal models.* Procs. of the int. conf. on conceptual modeling 2002

[28] Abrial J.R.: *The B book: assigning programs to meaning*, Cambridge university Press, 1995.